



05

## Paginación #1

### Transcripción

[00:00] Bien, ya estoy con los datos básicos para hacer mi listado y estoy consiguiendo traer los datos desde la base de datos, valga la redundancia, restringiendo los campos a los que el negocio me está pidiendo que son nombre, especialidad, documento e mail. Lo vemos aquí.

[00:19] Entonces la base ya está hecha. Ahora necesitamos aplicar las reglas de negocio ordenadas ascendentemente y paginado. Ahora yo voy a invertir esto para trabajarlo porque primero quiero paginarlo, quiero limitar los registros que llegan de la base de datos, antes de aplicar alguna estrategia para ordenar estos registros. ¿Cómo hacemos esto? Bueno, primero un poco de historia.

[00:50] Antes, antiguamente, hace unos cinco o siete años, para paginar, teníamos que enviar, digamos que los índices. Trabajábamos con las listas. La lista es esa estructura de datos que ven aquí y teníamos que enviar los índices de los elementos que quería mostrar. Por ejemplo, eran llamados offset y limit.

[01:09] Por ejemplo, si quería la página 1 era el offset 0 al límite 4, si es que quería solamente 5 registros en la primera página y así ir incrementando manualmente estos parámetros hasta conseguir la paginación. Spring se dio cuenta que esto obviamente era muy tedioso y es un caso de uso muy común.

[01:31] Por lo tanto, sprint decidió implementar y facilitarnos la vida en con una estrategia de paginación básicamente automática. ¿Cómo es esto? Bueno, lo que estoy retornando ahora aquí es una lista. Si me paro por aquí vemos que

es una lista lo que retorna, pero ya les dije que yo no quiero trabajar con listas, yo quiero trabajar con páginas.

[01:53] Entonces esto de aquí va a cambiar a Page, con mayúscula. Y escogemos Page Spring método main, y vemos que el código deja de compilar automáticamente. ¿Por qué? Porque lo que yo estoy dando aquí de respuesta es una lista y ahora debe ser una página. Bueno, un poco obvio.

[02:14] Ahora, segunda cosa que tenemos que hacer. Cuando nosotros trabajamos con páginas, necesitamos trabajar con un parámetro que llega del front end llamado Pageable, es este de aquí. Con Pageable, de data domain, lo vamos a llamar paginación.

[02:32] Este parámetro, que llega desde el cliente es opcional, va adentro del método findAll. Y ustedes me dirán: “pero antes findAll no usaba parámetros”. Okay, findAll es un método sobrecargado. Si vieron el curso de Java básico, van a ver que los métodos sobrecargados son método con el mismo nombre, pero que pueden aceptar diferentes parámetros.

[02:53] Entonces mandamos paginación como parámetro al método. Y, finalmente, como paginación va a retornar en sí un page, ya no necesitamos del API String de Java. Tampoco necesitamos mapearlo a una lista. De este modo tenemos la paginación y estamos mapeando cada elemento a lo que necesitamos de datosListadoMedico.

[03:22] Es decir, la paginación va a venir, Spring internamente va a ejecutar la misma query contra la base de datos, nos va a traer la entidad médico, un arreglo entidad médico, internamente Spring lo arregla bien bonito y lo transforma a datosListadoMédico, vamos a ver si es verdad todo esto, voy a darle clic y vemos ahora que el payload de respuesta es diferente. ¿Por qué?

[03:49] Porque antes teníamos directamente la lista de elementos y no teníamos nada más, ¿pero ahora qué tenemos? La lista está contenida dentro de un content. El content y también pageable, que es un body de la página en

sí. Por ejemplo, vemos que en sort, que es el atributo de ordenación, no tiene nada fuera de lo común, empty sort and sort no le hemos mandado ningún parámetro. Entonces, no hay ningún problema.

[04:24] Tiene el offset. Recuerdan que les comenté sobre offset, limit. Vemos claramente entonces que el pageNumber es la página 0, el pageSize 20, por defecto nos trae 20 registros por cada paginación que hagamos. No tenemos 20, porque no hay 20 registros en la base de datos.

[04:39] Pero vemos aquí toda la información con respecto a la página en sí, en pageable, en este objeto que estamos enviándole aquí, pageable, él nos dice toda la información con respecto a la página y esto es bueno para que el front end, nuestro cliente sepa en qué página estamos.

[04:59] De modo que dinámicamente, por ejemplo, vimos que aquí no le agregamos ningún body, ningún parámetro porque automáticamente esto ya tiene valores por defecto. Puede sonar un poco confuso ahora, lo vamos a ver con más detalle en el siguiente video.